UNITED STATES PATENT APPLICATION

for

METHOD FOR UTILIZING WAVEFORM RELAXATION

IN COMPUTER-BASED SIMULATION MODELS

Inventor:

Sunil C. Shah

Prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, Ca 90025-1026
(408) 720-8598

Attorney Docket No.: 001340.P006

METHOD FOR UTILIZING WAVEFORM RELAXATION
IN COMPUTER-BASED SIMULATION MODELS

FIELD OF THE INVENTION

5   The present invention is in the field of methodologies for engineering design activities, and more particularly in the field of methodologies for computationally intensive signal processing design or control system design activities.

BACKGROUND OF THE INVENTION

10   Design of new products is becoming an increasingly complex activity because of reliance on high performance features requiring signal processing and feedback control. Many industries today also rely on complex processes to produce a product. For example, the semiconductor industry uses extremely complicated processes to produce products that typically have very narrow

15   tolerances for final product characteristics. This situation presents a challenge for those designing products and control systems, in part because the design process is very computationally intensive. Similar challenges exist in any area where a complex product must be designed and its final characteristics tightly

20   controlled. Current methodologies for design of products and control systems are inadequate in several respects. For example in the area of control of manufacturing processes or control of product behavior, standard control methods are currently used. Many modern products and manufacturing processes, however, are too complex for standard control methods to

25   satisfactorily control them. Typical prior design methods are linear plans that do not provide alternatives required by the uncertainty of outcomes of computations and tests, and then permit planning based on resource utilization.

Also, these prior methods do not incorporate actual experienced results of process execution and adjust projections accordingly.

It is difficult, if not impossible, to achieve satisfactory results using prior methods. There are many problems in applying such methods to complicated

5 manufacturing processes or control of behavior of high performance products.

Prior methods implemented in design tools are not effective for several reasons. For example, prior design tools typically automate linear fragments of the design activity. Results of design steps are thus unknown or uncertain before the steps are actually carried out. For instance, compute time, computational

10 errors and exceptions, and results of physical tests cannot be known in advance to aid in decision making. These prior tools require a user to make a large number of complex decisions that depend on results of many previous steps. This is a disadvantage because the user must usually possess specific knowledge or skills in order to properly use the information gained from execution of

15 previous steps. It is a further disadvantage because intelligent decisions can only be made and incorporated after waiting for execution of design steps. No problem-specific guidance is available from prior tools for projecting with any accuracy what the results of design steps will be.

Most existing software design tools simply automate fragments of

20 standard design methods. In general, the tools are ineffective when applied to control of design and manufacturing processes. In the conventional paradigm, scientists, process experts, and control experts work within different domains with different tools. Scientists typically operate in the domain of physical models using a tool such as Fortran TWOPNT 106 (Grcar, J., The TWOPNT

25 Program for Boundary Value Problems, Sandia National Laboratories, SAND 91-8230, April, 1992). Process experts typically deal with the domain of process

monitoring using a tool such as Lab View® (available from National Instruments, Austin, Texas). Control experts use tools such as MATRIXX® (available from Integrated Systems, Inc., Sunnyvale, California), or MATLAB® (available from Mathworks, Inc., Natick, Massachusetts). Efficient control

5   requires an integration of information from each of in an easily usable format, which typically does not occur in current design tools.

A subsystem in a simulation model may consist of components such as an actuator in a reactor, the reactor vessel, reacting species, and the sensors. Because different physical phenomena may have to be simulated in different

10   subsystems, the use of specialized software for each subsystem provides significant benefits in development time and in running time. Different subsystem simulation software packages are developed by domain specialists using one or more of standard software packages. However, complete access to all internal state variables of each subsystem along with its data flow is needed

15   for integration of a complete and accurate system simulation.

The integration of information is prevented by the use of separate standard software packages such as those mentioned above. Without integrating the several software packages at the level of states of each subsystem, it is necessary to take very small steps during each iteration. The models in taking

20   small steps may thereby exchange data in near real time. Each model will calculate its data in the small steps and wait for the slowest model to complete. The models may then exchange their data as necessary. This method has the disadvantage of very slow operation.

In order to integrate the use of several standard software packages, it

25   would be necessary to have access to the source code of the packages or at least

have access to the data structures internal to the packages. It is seldom possible
to have the source code or the data structures internal to the software packages.

SUMMARY AND OBJECTS OF THE INVENTION

5   The present invention provides a method for designing high performance
products incorporating signal processing and feedback control. In one
embodiment, a block diagram may be used for a design cycle, for design
optimization, or for design estimation. The block diagram contains a set of
differential equations or difference equations that must be solved. The solution

10   of these sets of equations may be performed by commercially available software
tools. There will typically be more than one software tool used in a block
diagram system, with a software tool dedicated to each block. In order to utilize
the software tools without requiring access to source code or other descriptions
of the internal structure of the tools, the system is decomposed using the

15   technique of waveform relaxation. Once decomposed, the requirements for
interprocessor communication are minimized.

In one embodiment, the decomposition using waveform relaxation
operates directly to speed up the computations for the block diagram system,
whether it is in control system design, system parameter identification, or

20   product design cycle.

In another embodiment, the remaining interprocessor communications are
held pending until the end of each iteration's calculations in each block. In this
manner the software tools may be executed on independent multiple processors.

In another embodiment, additional low fidelity models are added to the

25   block diagram to accelerate the convergence and thus reduce the number of
successive iterations required for a given final accuracy.

In another embodiment, non-stationary waveform relaxation methods are used. In non-stationary methods, the model representing each block may vary with each successive iteration. The variation may take the form of successively increasing the complexity and therefore the fidelity of each block with each

5   successive iteration, starting with low fidelity models and finishing with high fidelity models. This advantageously gives fast convergence in early iterations and good accuracy in later iterations.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram of the logical structure of a design cycle according to one embodiment.

FIG. 2a illustrates a generic block diagram of waveform relaxation with strong coupling, according to one embodiment.

FIG. 2b illustrates a more specific example of waveform relaxation with strong coupling, according to one embodiment.

FIG. 3 is a graph showing errors in a Gauss-Jacobi waveform relaxation method, according to one embodiment.

FIG. 4a illustrates a generic block diagram illustrating an accelerated relaxation decomposition method, according to another embodiment.

FIG. 4b illustrates a more specific example of a block diagram illustrating an accelerated relaxation decomposition method, according to another embodiment.

FIG. 5 is a graph showing improved convergence in a waveform relaxation method, according to another embodiment.

DETAILED DESCRIPTION

In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details. In some instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical and other changes may be made without departing from the scope of the present invention.

Some portions of the detailed descriptions that follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of acts leading to a desired result. The acts are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to

the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

The present invention can be implemented by an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer, selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method. For example, any of the methods according to the present invention can be implemented in hard-wired circuitry, by programming a general purpose processor or by any combination of hardware and software. One of skill in the art will immediately appreciate that the invention can be practiced with computer system configurations other than those described below, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention can also be practiced in distributed computing environments where tasks are performed by remote processing

devices that are linked through a communications network. The required structure for a variety of these systems will appear from the description below.

The methods of the invention may be implemented using computer software. If written in a programming language conforming to a recognized standard, sequences of instructions designed to implement the methods can be compiled for execution on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, application...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or produce a result.

FIG. 1 is a diagram of the logical structure of an example design cycle 100 of one embodiment, which represents a design cycle for a process controller. Design cycle 100 includes major steps for design of a controller for a thermally activated process. In this embodiment, the thermally activated process is silicon oxide growth carried out in an oxidation furnace as in the semiconductor industry. Typically, during higher level iterations involving implementation and model refinement, the steps of design cycle 100 will be executed many times.

Referring to FIG. 1, nonlinear model identification step 102 comprises: 1) estimating uncertain parameters in a hypothesized set of ordinary nonlinear parameterized differential equations using test signals; 2) examining the validity of the hypothesized model; and 3) refining the structure and test trajectories to attain the desired degree of predictability from the nonlinear model.

Nonlinear simulation model step 104 comprises creating a computational simulation model consisting of a set of nonlinear ordinary differential equations such that the computational model can numerically integrate the equations given initial states, input signals and noise process trajectories. Closed loop simulation with a current controller step 106 includes a computational closed loop simulation model where a set of control loops with a currently existing controller are closed around the nonlinear simulation model of step 104.

Step 108 of linearizing around a steady-state includes computing a steady-state condition of the closed loop simulation with the current controller, then computing the linearized differential equations around the steady-state condition by numerical differentiation of nonlinear simulation model 104.

Step 110 of analyzing the high order model comprises computing the multivariable frequency responses, step responses, responses to stochastic noise processes and eigenvalues of the linearized differential equations from step 108. In step 110, a black-box linear model previously identified in step 112 is used.

Step 112, black-box linear model identification comprises estimation of parameters in a set of linear difference equations.

Model reduction step 116 comprises computing an approximate model of the linear high order model of step 108 with a fewer number of states than the high order model. This may also provide an estimate of errors due to the approximation.

Estimator controller synthesis step 118 comprises computing an estimator controller that optimizes certain properties, such as the H2, H-infinity or the Mu norm of the closed-loop system consisting of the reduced order model from step 116 and the estimator controller.

At step 120, the step of evaluating on the high order linear model includes computing multivariable frequency responses, step responses, responses to the stochastic noise processes, eigenvalues and measures of robustness for the closed system consisting of the high order model from step 108 and the estimator

5   controller from step 118.

Full order estimator controller synthesis step 114 comprises computing an estimator controller that optimizes certain properties, such as the H2, H-infinity or the Mu norm of the closed-loop system consisting of the high order model from step 108 and the estimator controller.

10   Controller reduction step 124 comprises computing an approximate estimator controller of the full order estimator controller of step 114 with fewer states than the full order estimator controller.

Step 126 of evaluating on the high order linear model includes computing multivariable frequency responses, step responses, responses to stochastic noise

15   processes, eigenvalues and measures of robustness for the closed system consisting of the high order model from step 108 and the reduced order model from step 124.

Step 122 of evaluating on the full nonlinear simulation includes: 1) replacing the current controller in the closed loop simulation of step 106 with the

20   estimator controller from step 118 or from step 124; 2) running non-linear closed-loop simulations for various initial values and perturbed model parameters.

Each step of the design cycle of FIG. 1 may potentially be executed by alternative methods. For example model reduction step 116 may be performed using the methods listed in Table 1.

25

<div align="center">Table 1</div>

Method	Applicability	Compute Costs	Reliability
Orthogonal Kalman Decomposition	Works without regard to stability. Very sensitive to scaling.	2.6 N3+6.5 N2	Poor
Balanced Realization	Requires stability. Gives H-infinity error bound	50 N3	Good
Schur Decomposition of Grammian Product	Requires stability, Gives H-infinity error bound	50 N3	Excellent
Hankel Norm Reduction	Requires stability. Tighter H-infinity error bounds than with Balanced Realization	50 (N3+ (N-1)3+ (N-2)3..)	Good
Modal Decomposition	Does not require stability. No bounds on H-infinity errors. Relies on time scale separation	20 N3	Good

In one embodiment of the present invention, a waveform relaxation

5   method is used to solve the simultaneous systems within multiple blocks.

Waveform relaxation is attractive because we can use separate solvers or

integrators for each decomposed system. There is no need to access internal code

information to integrate software modules. A strict input-output formalism can

be maintained. This reduces the engineering time spent in software integration

10   and permits solvers that are optimal for a given subsystem model structure.

Prior art methods for large-scale simulations such as the Krylov Space Solver,

used in DASPK, offer significant speed advantages but require exploiting

detained structure of the underlying simulation problem in constructing pre-

conditioners. Using waveform relaxation allows the retention of the speed advantages in simulating subsystems without the need for access to the internals of the simulation software.

An advantage of using waveform relaxation methods is that any continuation from previously computed waveforms may be computed more quickly. This allows rapid solutions for small, incremental changes in inputs, parameters, and initial conditions. In addition, multiple processor implementation is made easier because it is possible to wait for interprocessor communication until subsystem simulation of entire waveforms is completed.

Waveform relaxation may be performed using Gauss-Jacobi Theory. The Gauss-Jacobi theory utilizes matrix manipulations to convert variables for block diagram simulations. In general, the method is to decompose the original system of equations into subsystems. Each subsystem can be solved separately, assuming its input waveforms are known. During the next iteration, the new set of outputs is used to replace the older input waveforms. In the Gauss-Jacobi theory, the calculations for each iteration in each block may be advantageously completed prior to their being transmitted to the next block. In alternate embodiments, the Gauss-Seidel theory relaxation method may be used.

The conditions for convergence of the iterations, which are based on the application of a contraction mapping theorem, can be satisfied for certain decompositions of the original equations. Consider differential algebraic equations partitioned in the canonical form of Equations 1:

$$\dot{x}^k = f(x^k, x^{k-1}, \dot{x}^{k-1}, z^{k-1}, u) \, ; \qquad x^k(0) = x(0)$$
$$z^k = g(x^k, x^{k-1}, \dot{x}^{k-1}, z^{k-1}, u) \qquad \qquad \text{Equations 1}$$

where $x^k, z^k,$ and u are real vectors of length n, ℓ, and r, respectively.

If the inputs u are piece-wise continuous and the functions f and g are Lipschitz

5 continuous and globally contractive with respect to the iteration variables $(\dot{x}, z)$,

then the waveform relaxation iterations converge as $k \rightarrow \infty$ uniformly to those

shown in Equations 2:

$$\dot{\hat{x}} = f(\hat{x}, \hat{x}, \dot{\hat{x}}, \hat{z}, u) \; ; \qquad \hat{x}(0) = x(0)$$

10 $$\hat{z} = g(\hat{x}, \hat{x}, \dot{\hat{x}}, \hat{z}, u) \; ; \qquad t \in [0, T] \hspace{4cm} \text{Equations 2}$$

For differential equations in non-normal form, diagonal dominance conditions

and Lipschitz continuity are sufficient.

As an example, for open-loop simulation of thermal conduction-radiation

15 models, the structure of the equations resulting from finite difference

approximation of transient energy balance equations is given by Equation 3:

$$C(x)\frac{dx}{dt} = f(x, u(t)) \qquad , x(0) = x0 \hspace{3cm} \text{Equation 3}$$

20 Diagonal dominance of the thermal mass matrix C(x) is trivially satisfied in

thermal conduction-radiation models. This property allows, if desired, breaking

up of the open-loop simulation model into subsystems as fine as one node.

When dynamic control feedback equations are augmented to these non-

normal differential equations, the resulting equations have the structure shown

25 in Equations 4 below:

$$C(x)\frac{dx}{dt} = f(x,x_c)$$
, x(0) = x0

$$\frac{dx_c}{dt} = g(x_c, x, r(t))$$
, xc(0) = xco   Equations 4

Again, diagonal dominance of the matrix $\begin{bmatrix} C(x) & 0 \\ \hline 0 & I \end{bmatrix}$ for the resulting non-normal

equations is immediate from the structure. This allows the combined closed-loop

system to be broken up into first order equations, and allows waveform

relaxation to be performed with provable convergence properties. If discrete-

time controller equations are used in place of continuous controller equations,

the convergence properties still remain valid.

Waveform relaxation theory decomposition of the differential equations

or difference equations within a computationally complex simulation may be

used to speed up the computations.

FIG. 2a and FIG. 2b are block diagrams showing a waveform relaxation

method with strong coupling, according to one embodiment. FIG. 2a illustrates a

generic block diagram of waveform relaxation with strong coupling and FIG. 2b

illustrates a more specific example of waveform relaxation with strong coupling.

It should be noted that FIG. 2a represents a generic block diagram

illustrating a waveform relaxation method in a system with strong coupling. The

generic illustration of FIG. 2a is used herein to demonstrate the breadth of scope

of the present invention. FIG. 2b, discussed below, is a more specific example of

a block diagram illustrating a waveform relaxation method in a system with

strong coupling. FIG. 2b is not meant to be limiting on the scope of the present

invention and is meant merely to be exemplary.

In the more general embodiment of FIG. 2a, the present invention is used

in order to enable faster convergence in computationally complex systems.

Waveform relaxation decomposition in computationally complex systems is performed by first regularizing the algebraic loops of the system by introducing $\dfrac{1}{\tau s + 1}$ into the algebraic loop fast dynamics such as , where τ is small. The system is then decomposed to the lowest level of available inputs and outputs

5   from the existing software modules of the system.

In typical block-diagram forms used in control models, each block is in the normal form of Equations 5:

$$\dot{x}_i = f_i(x_i, u_i)$$
$$y_i = g_i(x_i, u_i)$$

10   Equations 5

In Equations 5, the fi are the inputs and the gi are the outputs of a given state of a block.

The connections among the blocks are given by a sparse interconnection

15   matrix Econnect in Equation 6 below:

$$\begin{bmatrix} u_1 \\ . \\ . \\ . \\ u_n \end{bmatrix} = E_{connect} \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_n \end{bmatrix} + r_{external}$$

Equation 6

The sparse interconnection matrix Econnect of Equation 6 is utilized to allow the

20   use of waveform relaxation. The presence of direct feed-through terms in function g (i.e. the partial derivatives of g with respect to u being non-zero) is critical in determining the ordering of the blocks in simulations. Most block diagrams do not have algebraic loops or can be regularized so as not to have

algebraic loops. Consequently, a particular sequence of block output evaluation allows the solution of all block inputs using the states and the external inputs. The sequencing results in normal form equations for the interconnected system. The reordering of blocks creates a thread or chain of computations terminating at blocks without direct feed-through terms, or at blocks with external outputs.

A natural decomposition for waveform relaxation of block diagrams may be based on inputs and outputs of subsystems. Blocks within a chain should be solved sequentially. Waveform relaxation of such subsystems only requires Lipschitz continuity of subsystem equations.

The chain of computations for the block diagram are determined based upon the dependency of the direct feed-through terms of the system. The chains of computation that have direct terms should remain as complete chains and should not be broken down. The resulting block diagram consists of a number of feedback loops, each loop consisting of a computationally complex feedforward part 250 and a feedback part 260, as shown in FIG. 2a.

In the example of FIG. 2a, a system 290 is shown as having a computationally complex feedforward part (feedforward part) 250 that is coupled to the remainder of the system 290 (remaining feedback system 260). Feedforward part 250 produces a signal 251 that is a set of output variables which are converted using waveform relaxation into relaxation variables 270 that are used by the remaining feedback system 260. Remaining feedback system 260 produces a signal 261 that may also be a set of output variables and which may also be converted into relaxation variables 280 that are used by feedforward part 250.

In the FIG. 2b embodiment, a control simulation 200 is shown. In alternate embodiments the blocks could describe closed-loop simulation computations in

system identification (the derivation of coefficients or other parameter estimate, or state estimate in the case of stochastic processes) or a product design cycle. The control simulation 200 contains a high fidelity plant model 224 and a low order controller 234. High fidelity plant model 224 is necessary for system

5   accuracy. The feedback generated by low order controller 234 is sufficient for system stability. The high fidelity plant model 224 is one example of a class of objects called feed-forward elements. Similarly, the low order controller 234 is one example of a class of objects called feedback elements.

High fidelity plant model 224 operates on a signal generated from a

10   reference input signal 210 and an error signal 238 which is combined at mixer 214. The output of mixer 214 is a set of non-relaxed variables which are converted by waveform relaxation into relaxation variables u 220. Relaxation variables u 220 serve as the input of high fidelity plant model 224 and produce an output 228. The output 228 is again transformed by waveform relaxation into

15   relaxation variables y 230. Relaxation variables y 230 are operated upon by low order controller 234 to produce error signal 238. The transformation into waveform relaxation variables u 220 or relaxation variables y 230 may be accomplished using Gauss-Jacobi theory, as discussed above.

20   FIG. 3 is a graph showing errors in a Gauss-Jacobi waveform relaxation method, according to one embodiment. Global convergence of Gauss-Seidel and Gauss-Jacobi waveform relaxation using the contraction mapping theorem leads to convergence in an exponentially weighted norm. However, uniform convergence over the simulation time interval with an unweighted norm is not

25   available unless certain stronger conditions of diagonally dominant non-negative

monotonicity hold. Tight coupling among decomposed subsystems slows down convergence in MOS circuit simulations.

For closed-loop control simulations, it may be shown that, although waveform relaxation solutions do converge, unmodified waveform relaxation may require many iterations. This slow speed of convergence is a direct result of strong coupling between the plant model and the controller, as shown in FIG. 2b above. Strong coupling in a feedback loop occurs when loop-gain of the particular feedback loop exceeds unity. The graph of FIG. 3 shows results of Gauss-Jacobi iterations applied to a similar control system simulation. The simulations indicate slow convergence of the output waveforms.

One method to alleviate this problem is to run simulations over shorter intervals and use partial waveform convergence. However, having a greater number of time intervals increases communication overhead. In a case with a great number of very short simulation intervals, any speed advantages of waveform relaxation may be lost.

Because normal convergence is slow enough to potentially mask the performance improvements of waveform relaxation, a fast convergence technique is desirable.

FIG. 4a and FIG. 4b are block diagrams showing an improved waveform relaxation method with weaker coupling, according to another embodiment. FIG. 4a illustrates a generic block diagram for accelerated relaxation decomposition and FIG. 4b illustrates a more specific example of a block diagram for accelerated relaxation decomposition.

It should be noted that FIG. 4a represents a generic block diagram illustrating an accelerated relaxation decomposition method. The generic

illustration of FIG. 4a is used herein to demonstrate the breadth of scope of the present invention. FIG. 4b, discussed below, is a more specific example of a block diagram illustrating an accelerated relaxation decomposition method. FIG. 4b is not meant to be limiting on the scope of the present invention and is meant

5   merely to be exemplary.

In FIG. 4a, a system 490 is shown as having a computationally complex feedforward part (feedforward part) 460, reduced model feedback parts 465, 466, which are all coupled to the remainder of the system 490 (remaining feedback system 470). Feedforward part 460 produces a signal 461 that is a set of output

10   variables, reduced model feedforward parts 465, 466 produce signals 468, 469 that are approximations of signal 461. One or more of the signals 461, 468, and/or 469 may then be converted using waveform relaxation into relaxation variables 480 that are used by the remaining feedback system 470. Remaining feedback system 470 produces a signal 471 that is a set of output variables which

15   are converted using waveform relaxation to into relaxation variables 485 that may be used by feedforward part 260 and reduced model feedforward parts 465, 466.

The feedback loops of the block diagram having strong coupling are identified. Feedback loops with strong coupling may simply be identified

20   because certain feedback loops in control are expected to have strong coupling or can also be identified by discovering slow convergence under feedback in a standard relaxation simulation (as described above with regard to FIG. 2b). In order to reduce the strong coupling for a given feedback loop the computationally complex feedforward part 460 may be approximated by a

25   simpler model, for example, by a reduced model feedforward part P' 465, 466 as described above, to create new relaxation variables 480 and 485.

In the FIG. 4b embodiment, high fidelity plant model 422 is computationally complex and so control simulation 400 exploits reduced plant models 424, 444. In this exemplary method, fast convergence arises through the use of low fidelity models 424, 444. In other words, the computationally complex plant 422 is approximated by using the simpler reduced plant models 424, 444 in order to reduce strong coupling. The approximation of complex plant 422 may be performed using model reduction techniques and/or system identification techniques. Reduced plant models 424, 444 are examples of low fidelity models.

The small errors between high fidelity plant model 422 and reduced plant models 424, 444 are only weakly coupled with the reduced closed-loop simulation. For this reason the convergence rate is greatly enhanced. Standard variables u 418 are presented to reduced plant model 444 and relaxation variables u 420 are presented to high fidelity plant model 422 and reduced plant model 424. The error signal produced by mixer 430 is used to create another set of relaxation variables 440. These relaxation variables 440 are combined by mixer 446 with the output of reduced plant model 444 to yield error signal 450. The low order controller 452 feeds back error signal 450 in the operation of control simulation 400.

FIG. 5 is a graph showing improved convergence in a waveform relaxation method, according to another embodiment. In the FIG. 5 graph, the convergence of the FIG. 2b control system 200 (standard) is contrasted with the convergence of the FIG. 4b control system 400 (accelerated). Weak coupling associated with the reduced plant models 424, 444 gives faster convergence for the accelerated control system 400.

An additional benefit of the FIG. 4a and FIG. 4b decomposition methods is that the weak coupling with the waveform relaxation allows the use of multiple processors and simultaneous modeling. The solution of each block's differential equations or difference equations only weakly requires interprocessor communications. In one embodiment, the interprocessor communications may be delayed until each block's calculations are finished. In this manner the calculations for each block may be calculated separately, and therefore may be calculated on multiple processors. The calculations for the entire control system may therefore take place simultaneously.

The methods described above are called stationary waveform relaxation. The methods are stationary in that the model equations in each block do not change from one iteration to the next. In contrast, non-stationary waveform relaxation allows the model equations in each block to vary in time. Non-stationary waveform relaxation allows for further improvements in convergence speed and accuracy.

The method of non-stationary waveform relaxation operates by increasing the complexity and therefore fidelity of models with each successive iteration. In early iterations, low fidelity models are used, which contribute to the speed of convergence. In subsequent iterations, successively higher fidelity models are used. By using successively higher fidelity models, non-stationary waveform relaxation gives the benefits of fast convergence during early iterative steps in conjunction with high fidelity and small errors in the final iterations.

Examples of models with varying fidelity include gain scheduled time series models, nonlinear black box dynamic models (including neural network structures), and nonlinear physical models with differing numbers of states.

System identification refers to the process of determining the coefficients or other parameters of the differential equations or difference equations of a block. In the case of stochastic processes, system identification may include determining state estimates. It is noteworthy that empirical models of increasing

5   complexity and accuracy are obtained as a byproduct of the system identification process.

Another application of non-stationary waveform relaxation is in evaluating sensitivity functions for numerical trajectory optimization and system identification. One application of trajectory optimization is described in Shah et

10   al., Patent No. 5,517,594, Figure 7, assigned to the assignee herein. Numerical derivatives of output trajectories may be evaluated with respect to uncertain physical parameters for system identification. Computation of numerical derivatives of output trajectories for system identification is described in System Identification, Theory for the User, Ljung, 1987, Prentice Hall, pages 282-287. To

15   evaluate perturbed simulation trajectories, the unperturbed trajectory can be used as a starting trajectory in a waveform relaxation approach.

The present invention has been described in terms of specific exemplary embodiments. For example, various aspects of an embodiment that is used for designing a control system for a thermally activated process have been described

20   as examples. Other embodiments of the present invention are used for design of other processes with various modifications and alterations that might be made by those skilled in the art without departing from the spirit and scope of the invention as set forth in the following claims.